

«Национальный исследовательский университет
«Высшая школа экономики»
Лицей

Индивидуальная выпускная работа

ИТ-ПРОЕКТ: RPF

<https://github.com/YakovIVR/RPF/>

Выполнил(а): Дергачев Яков Константинович

Группа: 11u3

Москва – 2023

У меня небольшой опыт в разработке. Когда-то я делал различные мини игры и к ним механики и шаблоны. Я ни разу не работал над полноценными проектами и к тому же мне нравится программировать и создавать что-то новое. Благодаря этим факторам, у меня появилось желание создать что-то значительное, похожее на настоящий проект. И поэтому я выбрал IT-проект.

Все началось с того, что я понял, что надо уметь готовить хоть что-то. Я заметил, что с каждым днем / годом все больше людей предпочитают ужинать в ресторанах, кафе и других заведениях, вместо того чтобы готовить еду дома. Это приводит к проблеме: многие забывают, как готовить домашние блюда, и к тому же, родители часто не учат этому своих детей. И в связи с этим, я решил создать приложение, которое будет помогать в приготовлении пищи.

Мое приложение предоставляет обширный ряд функций. Я расскажу только о некоторых из них. Мой продукт позволяет просматривать рецепты как в online режиме, так и в offline, но в offline режиме при условии, что пользователь сохранил какое-нибудь количество рецептов. Это может быть очень удобно если, например, в рационе только определенные блюда, то можно скачать рецепты / способы приготовления этих блюд и не тратить гигабайты мобильного или домашнего интернет на поиск этих же блюд. Еще это может быть удобно, если надо экономить интернет, для этого есть много причин, не будем их перечислять, можно так же просто скачать определенные рецепты и просматривать их, когда угодно. Так же приложение предоставляет возможность самому создавать рецепты. При этом не обязательно быть поваром, если знаешь рецепты, то почему бы их не создать. Так же можно создать объявление к повару. Это помогает решать проблему, когда не умеешь готовить или просто не хочешь. Или же наоборот, приложение предоставляет возможность повару найти клиентов. Повар в данном случае – человек, который умеет хорошо готовить требуемую еду. Еще мой продукт позволяет удалять созданные рецепты, фильтровать рецепты по множествам критериев, смотреть рейтинг у повара и многое другое. Эти все функций таким или иным способом позволяют решить поставленные проблемы.

Мне удалось реализовать все пользовательские сценарии. Первый сценарий, который я реализовал был сценарий авторизации пользователя. Моя реализация этой механики ничем не выделяется от остальных реализаций и методов. Можно зарегистрироваться если нет аккаунта, можно войти уже в существующий аккаунт или же, восстановить пароль по почте. Вторым сценарием была возможность смотреть рецепт как с интернетом, так и без него. Можно смотреть информацию о рецепте и готовить его по шагам, я постарался сделать это максимально удобным для пользователя. Я уже достаточно подробно описал сценарии про заказ клиента и заказ повара в предыдущем абзаце. В сценарии, когда клиент хочет заказать повара, чтобы это сделать надо перейти на страницу «Поиск повара» и выбрать подходящего повара. А в противоположном этому сценарию, где повар хочет найти клиентов, то он может зайти на главную страницу для поваров, где есть все запросы к поварам и откликнуться на любой запрос. Если же вы хотите по какой-либо причине удалить написанный вами рецепт, то нужно отфильтровать все рецепты по ключу «Мои рецепты» и удалить нужный рецепт. Если у вас есть какие-нибудь продукты в холодильнике, то вы можете выбрать сортировку по ключу «Рецепты, которые я могу приготовить из холодильника» и высветятся все рецепты, в которых есть хотя бы один ингредиент из вашего холодильника. Так же мой продукт предоставляет возможность редактировать рецепт. Для этого нужно повторить сценарий для удаления рецепта, но не удалять. Рядом с кнопкой «Удалить рецепт» будет кнопка «Редактировать рецепт». По ее нажатии открывается меню редактирования рецепта. пользователь может изменить в рецепте все, что угодно: от главной фотографии, до количества шагов и количества таймеров в этих шагах. Еще в приложении можно обновить информацию о поваре. Для этого нужно войти в аккаунт повара, потом в личный кабинет повара и нажать кнопку «Редактировать профиль». После этого откроется меню редактирования профиля, где так же можно отредактировать что угодно, исключая имя: фотографию повара, краткую информацию о поваре, расположение повара и его номер телефона. После всех изменений надо нажать кнопку «Сохранить» и изменения сохранятся.

Теперь я расскажу о подробностях моего бэкенда. Выбирал я не долго, использовал то, что знаю. Поэтому мой выбор пал на MySQL. Взаимодействую я с ним, если не из кода, то с помощью `phpmyadmin`. Ниже прикреплена структура каждой таблицы.

<div> <div>v</div> <div>⚙</div> <div>cshse_22 users</div> <div>id : int(11)</div> <div>username : varchar(50)</div> <div>password : varchar(100)</div> <div>email : varchar(2000)</div> </div>	<div> <div>v</div> <div>⚙</div> <div>cshse_22 requests_to_shief</div> <div>id : int(11)</div> <div>username : varchar(1000)</div> <div>request_name : text</div> <div>metro : varchar(100)</div> <div>price : int(11)</div> <div>chiefs : text</div> </div>
<div> <div>v</div> <div>⚙</div> <div>cshse_22 recipes</div> <div>id : int(11)</div> <div>username : varchar(11)</div> <div>picture : text</div> <div>recipe_name : varchar(100)</div> <div>sum_score : float(3,1)</div> <div>views : int(11)</div> <div>score_count : int(11)</div> <div>score : float(3,1)</div> </div>	<div> <div>v</div> <div>⚙</div> <div>cshse_22 chiefs</div> <div>id : int(11)</div> <div>image_path : text</div> <div>chief_name : varchar(1000)</div> <div>views : int(11)</div> <div>score : float(3,1)</div> <div>phone_number : varchar(19)</div> <div>metro : varchar(100)</div> <div>short_about_yourself : text</div> <div>responded : text</div> <div>is_add : int(1)</div> </div>
<div> <div>v</div> <div>⚙</div> <div>cshse_22 about_recipe</div> <div>id : int(11)</div> <div>recipes_id : int(11)</div> <div>username : varchar(11)</div> <div>recipe_name : varchar(100)</div> <div>description : text</div> <div>ingredients : text</div> <div>steps_descriptions : text</div> <div>links_to_pictures : text</div> <div>timers : varchar(10000)</div> </div>	<div> <div>v</div> <div>⚙</div> <div>cshse_22 ingredients</div> <div>id : int(11)</div> <div>ingredient : varchar(100)</div> <div>recipes_id : text</div> </div>
	<div> <div>v</div> <div>⚙</div> <div>cshse_22 fridges</div> <div>id : int(11)</div> <div>username : varchar(50)</div> <div>ingredients : text</div> </div>

Я старался сделать максимально оптимальное количество таблиц и полей в них. Я считаю, что с этой задачей я справился. Теперь поговорим о связях в этих таблицах, то есть какая таблица связана с какой и каким полем. Например, таблица **recipes** связана с таблицей **users** полем *username*, а таблица **about_recipe** связана с таблицей **recipes** полем *recipes_id*. В свою очередь таблица **chiefs** связана с таблицей **users** полем *chief_name*, а таблица **fridges** связана с таблицей

users полем *username*, когда таблица **requests_to_shief** связана с таблицей **users** полем *username*, Таблица **ingredients** связана с таблицей **recipes** полем *recipes_id*.

Перед началом разработки у меня встал вопрос: «На чем писать и что использовать?» После некоторого количества времени над рассуждениями и остановился на следующем. Для клиентской части я использовал Unity и язык программирования C#, а для серверной части я использовал фреймворк Flask и язык программирования Python. Использовал я эти средства потому, что мне они больше всех подходили. Как я в начале говорил, раньше я делал небольшие шаблоны и механики и делал я их как раз-таки на Unity. Python я знал на максимально базовом уровне, но мне этого вполне хватило. Для дизайна я использовал Adobe Illustrator.

Этапы разработки были следующими: 1) Разработка пользовательских сценариев. 2) Проектирование интерфейса. 3) Заполнение базы данных. 4) Подключение БД. 5) Создание и подключение сервера. 6) Запуск Android версий продукта. 7) Тестирование, отладка. 8) Подготовка проекта к защите. Во все, прошедшие на данный момент дедлайны, мне удалось уложиться. Можно уделить внимание сценариям 1,2,3. Я пойду по порядку. Разработка пользовательских сценариев проходила следующим образом: я написал первую версию пользовательских сценариев, после этого реализовал почти все функции, а потом перестроил пользовательские сценарии под написанный продукт. Проектирование интерфейса. Это, наверное, была моя самая нелюбимая часть разработки потому, что я не дизайнер и никогда до этого не создавал дизайн для приложения. Под дизайном я имею ввиду весь внешний интерфейс. Создавал я его относительно всего проекта очень долго, мне надо было учесть наперед всевозможные действия пользователя и «нарисовать» их. Но я справился с этим. Заполнение базы данных. Я ее заполнял с помощью скрипта и предварительно подготовленных данных.

Анализируя проделанную работу хочется выделить несколько моментов: 1) Какие проблемы возникли у меня в процессе работы над продуктом и как я их решил. 2) Возможные пути развития продукта, что можно улучшить / добавить.

3) Какие компетентности я приобрел в процессе работы над продуктом и как я их смогу использовать в будущем. 4) Какие были риски в начале работы над проектом и осуществились ли они. Говоря о проблемах могу отметить, что их было довольно много. Так как до этого, как я говорил ранее, особого опыта в разработке не было. Например, одной из проблем было сохранение изображений на сервере. Изображения, которые весили около 50 кб, сохранялись с весом 2000 кб. Я долго думал почему это может быть и в итоге все оказалось не так очевидно лично для меня. Проблема заключалась в том, что я сохранял их в формате png, а надо было в формате jpg. Я в тот момент даже не догадывался до этого. Еще была проблема с сохранением изображения, которое было получено с камеры телефона. Но перед объяснением этой ошибки, надо объяснить, как я сохранял изображения. Я брал имя изображения и сохранял на сервере, так как думал, что имя изображения всегда уникально. Теперь вернемся к изображению с камеры. Изображения, полученные с камеры, имели одинаковое название. И как можно догадаться, если на сервере существует уже изображение с камеры, то оно просто перезаписывалось. Я решил эту проблему следующим образом. Я создавал уникальный идентификатор для каждого изображения и сохранял. Еще была проблема с плохой оптимизацией – приложение зависало на несколько секунд при обновлении рецепта. Проблема оказалась уже знакомой. Это было тоже из-за конвертирования в png, а не в jpg. Эта операция занимала намного больше времени, чем было необходимо. Теперь рассмотрим пункт 2. В данный момент я не планирую развивать данный продукт. Но его можно доработать, оптимизировать и адаптировать под все устройства android и выпустить в Google Play. Говоря о пункте 3, я приобрел такие компетентности и навыки как: самостоятельность, ответственность, time management, способность быстрее находить решение ошибок. Это сильно поможет мне в будущем при разработке новых проектов ну и в целом в жизни. Теперь обсудим пункт 4. У меня было несколько рисков, первый из них – это сделать плохую оптимизацию. У меня получилось избежать этот риск в какой-то степени, оптимизация в приложении лучше, чем я думал, но можно еще лучше. Следующий риск – это сделать

приложение слишком тяжелым (>500МБ). Этот риск тоже получилось избежать, приложение весит около 40 мб при загрузке и около 100 мб лично у меня на телефоне. И третий риск заключался в том, что я сделал бы неприятный интерфейс. Я считаю, что у меня получился хороший и приятный интерфейс. Он не режет глаз, и приложением удобно и приятно пользоваться.

В заключении я хочу сказать, что проделал для себя огромную работу. Я делал этот проект около двух месяцев и для меня это было тяжело, но при этом мне не хотелось сдаваться и поменять область. Я получил очень много новых для себя навыков, которые обязательно пригодятся в будущем.